TU WIEN Informatics

Audio Spatialization of Process-Windows

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Stephan Sgarz
Registration Number 11910593

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Mag.rer.nat. Dr.techn. Hannes Kaufmann

Vienna, February 3, 2025

Stephan Sgarz	Hannes Kaufmann

Erklärung zur Verfassung der Arbeit

Stephan Sgarz

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 3. Februar 2025

Stephan Sgarz

Abstract

Virtual, mixed and augmented reality all offer unique approaches and experiences in a digital space. Apart from entertainment, sectors like health and education long used immersive technologies to their advantage. However, the private use of these systems for common digital tasks has seen minimal support, since the required hardware tends to be cost intensive, cumbersome, or too complex to benefit simple procedures. This thesis aims to create a prototype that has minimal technical requirements and increases the immersion of general computer usage through binauralization. The program registers incoming audio sources, that are represented through a window and then transforms the position of the audio to match its perceived location on screen, creating an immersive three-dimensional audio field, while using regular two-dimensional screens. This is done through the application of virtual audio cables that separate audio objects from the standard output and delegate them to a Digital Audio Workstation. There, the audio gets binauralized to match the location of the process window and is then returned to the standard speaker output. This allows the audible position to be changed in real time, should the window be moved on screen. The code base for the prototype as well as example videos are accessible on `https://www.vr.tuwien.ac.at/projects/audio-spatialization-of-windows/`.

Contents

Abstract **v**

Contents **vii**

1 Introduction **1**
1.1 Goal of the Thesis and Idea 2

2 Related Work **5**
2.1 Spatial Virtualization 5
2.2 Immersion 6
2.3 Digital Audio Rendering 7

3 Methodology **13**
3.1 System Design 14
3.2 Choice Of Technology 18

4 Implementation and Use **21**
4.1 Used Assets, Packages and Environments 21
4.2 Reaper Project Setup 22
4.3 Interaction Flow 25
4.4 Limitations and Known Issues 25

5 Results and Outlook **27**
5.1 Future Improvements 27
5.2 Extended Applications 29
5.3 Conclusion 30

List of Figures **31**

Bibliography **33**

CHAPTER 1

Introduction

In recent years, immersive technologies have been steadily improved upon and are a major focus of research in the digital field. Immersive technologies, including VR (Virtual Reality), MR (Mixed Reality) and AR (Augmented Reality), try to minimize the sensory boundary from a virtual world to the physical one. The most prominent sensory outputs that are used in immersive systems are visual, aural and haptic stimuli. These technologies can range greatly in the amount of immersion they provide. [SP18]

Full VR head-mounts along with haptic sensory input nearly completely immerse the user in a virtual space, while simple adjustments to a visual or audio output apply comparatively small amounts of immersion to the user. The degree of immersion depends heavily on the specific use case and comes with different benefits and drawbacks.[SP22]

According to Ayoung Suh's and Jane Prophets research, this more immersive interaction with digital programs has shown to reap many beneficial effects such as, enhanced learning experiences, fostered participation in collaborative activity, as well as increased creativity and engagement.[SP18, p.1] However, Polina Häfners findings on the downsides and limitations of immersive technologies in learning environments show, that currently used hardware devices and more immersive systems may have several negative effects on the user. Among other remarks the studies showed that users were unhappy with the feel of some of the gear, like for example the weight of the headset, the cost of hardware tools as well as a lack of social presence and attention tunneling.[Hä20, p.158]

Focusing on immersive audio systems, there have been a great number of inventions and improvements in recent years such as professional cinema systems, that create immersive surround sound and MR or AR devices and appliances, for consumers that aim to immerse the viewer in games or learning tasks. There are two main ways to approach a heightened aural immersion in users. Either put the user in a space, where speakers cover most of the environment to create a sound field in which the user can move freely, or, using primarily headphones, manipulate the sound to create the illusion of it coming

from specific direction in the virtual space. This thesis focuses on the usage of the latter technology, specifically binauralization. An algorithm, the Head Related Transfer Function (HRTF), converts stereo or mono sound into a stereo format, that mimics the distortion of audio as the outer ear of a human would, making it appear to come from a specific direction anywhere around the user. Nowadays binaural Audio is used in Music, Films and Games among other applications and either follows the head movement of the user via a motion sensor or uses the position of a camera or character as the center point. [Sun21]

1.1 Goal of the Thesis and Idea

The goal of this thesis is to create a simple and easy to use system that improves audio immersion on Windows PCs, to enhance focus and learning experiences as well as heighten creativity and attention, while keeping the cost as well as required hard- and software minimal to reduce drawbacks. The application is made to be run in the background, requiring minimal input, and make aural location-transformations in real time.

Over the years all operation systems improved the visual look and feel of their desktops designs heavily, to increase usability and workflow. The two-dimensional nature of computer screens limits the amount of visual immersion heavily. There are already approaches to transform the desktop into a VR environment like the project from Virtual Desktop, Inc. [VD]. However, acceptance, cost and ease of use hinder the general transition to Virtual Reality in computer usage. However, the aural aspect of immersion regarding the use of computers shows a lot of untapped potential to further increase usability, since it is not necessarily bound to additional hardware and presents less of a change for the user. Most speakers or headphones already posses the necessary technology features to make sound appear more realistic and multidimensional. Various software processes are able to create three-dimensional sound fields, that yield a high amount of immersion.

The approach to convert standard desktop audio into a virtual three-dimensional space, is similar to the idea behind curved monitors. According to a study by Kyung and Park, curved computer screens appear to have a positive impact on accuracy, speed, and reduced fatigue of users compared to flat screens.[KG21] For this project, the normally flat and two-dimensional grid of a computer monitor is mapped around the users head in a hemisphere. Then, sound producing active process-windows are identified, streamed through a separate audio channel, and transformed using the binaural method as well as the positional data of the process-window. Meaning that, if a process-window on the left side of the screen produces sound, this signal should be heard as if it came from the left side of the users head.

As previously mentioned, binaural audio uses the Head Related Transfer Function (HRTF) to mimic the way human ears warp sound from specific directions to create a sense of orientation. [TT23] To make these transformations in real time and include currently running and newly started audio processes, a loop is constantly gathering window- and

audio data. These informations are then fed into a digital audio workstation to transform specific windowed audio processes. If the user drags an active audio window, from one end of the screen to another, the sound should follow that movement. With this, higher immersion on regular screens could be achieved. Simultaneously, running audio-producing windows could be more easily separated by the brain to enhance overall usability and orientation.

CHAPTER 2

Related Work

To realize an audio spatialization of process windows, several techniques and aspects need to be analyzed. First, we will look at the technical concept of virtualization, including virtual space, the environment and the user. Next, the principle of immersion presents a crucial role in allowing the user to improve their digital workflow and focus while using digital programs. On a technical level we will focus on the most recent immersive audio manipulation approaches, that create a virtual three-dimensional audio environment around one or more users. Throughout this section, related projects and ideas will cover the research and state of the art in the aforementioned key aspects.

2.1 Spatial Virtualization

The virtualization of space refers to constructing digital surroundings in which the users can navigate and act through sensory in- and output. Those spaces can either be replications of real-world areas, invented or simulated environments, or be abstract places.

Through virtual space, users can work and orientate themselves quickly using their knowledge of the material world, all while utilizing the benefits of digital technologies, like safety and comfort. Key technologies in creating virtual environments are 3D Modeling and Simulation technologies, spatial computing as well as haptic and sensory in- and output simulations. Using these principles, various processes and jobs are virtualized to either practice for real world scenarios or to enhance already digital tasks. [SPW21] Even in medical fields like neuroscience, virtual environments are used to treat patients with neurological afflictions to heighten or train their spatial awareness.[CNP23] For this thesis, however, we will focus mainly on sensory in- and output simulation to create immersive virtual space.

One of the largest fields for sensory in- and output simulation is motion tracking. Hereby user's limb-, head- or whole body-movements get registered by a sensor and transformed into virtual movement inputs. Head tracking technologies, for example, transfer the movement of the users head into a respective change of the users virtual point of view. To gain this transformation data, sensors are placed on the users' head that track accelerations and rotational velocities. [FMMH19, p.3]

Lastly even in more common settings like a standard digital game played on mouse and keyboard, improving the sensory in- and output can lead to creating more realistic virtual spaces. Binaural audio for example has slowly worked itself up in the industry offering a more realistic audio representation of space than stereo sounds. Binaural audio mimics the way humans naturally experience sound through headphones and allows the users to accurately spatially locate digital objects. Games like Overwatch (2016) and Rainbow six: siege (2015), use the advantages of binaural audio to make their games space not only more realistic, but to improve the immersion of their players.[BM23, p.19]

Consoles like the Nintendo WII, constantly experiment on and improve their controller vibration technologies as well as alternative input options like motion sensors. Developers of games or applications then have the opportunity to include new and more immersive elements like haptic input. These attempts aim to immerse users further into their created virtual space, quicken reaction times or heighten precision in localization and orientation.[Blo18]

2.2 Immersion

Digital immersion refers to the extent to which a user becomes absorbed in a digital environment, to the point where the boundaries between the user and the digital world become blurred. Sarvesh Agrawal at al. define immersion in their report as, a phenomenon experienced by an individual when they are in a state of deep mental involvement in which their cognitive processes (with or without sensory stimulation) cause a shift in their attentional state such that one may experience disassociation from the awareness of the physical world.[ASB+19, p.5] Immersion encompasses a range of experiences from high-level immersion, such as Virtual Reality (VR) and Augmented Reality (AR)) to low-level immersion like using a computer or smartphone.

While high-level immersion involves sophisticated technologies to heavily involve the user in the digital context, low-level immersion can occur through more ubiquitous digital interactions, such as browsing the internet, playing video games, or using mobile applications. The degree of immersion is influenced by factors such as the user's engagement, the design of the digital environment as well as the hardware supporting it, and the sensory and haptic in- and output complexity.[Sla18] This project will try to enhance the low level immersion of working on a digital desktop, without completely changing the workflow and without adding tools or hardware not accessible to the general user.

Both low- and high-level immersion can enhance user engagement. For example, in

digital marketing, immersive experiences can increase consumer engagement, leading to higher conversion rates and greater emotional attachment of customers. [TH24] In entertainment, immersive games and narratives keep users engaged for extended periods, enhancing their overall experience. However, immersive systems slowly expand from their primarily hedonistic usage and are beginning to be applied in utilitarian functionalities. In professional settings, for example, immersive tools can improve productivity by creating focused work environments. Tools like virtual desktops or more complex systems like entire VR workspaces allow for more immersive and engaging work processes, aiming to improve users' concentration and collaboration, especially in remote work settings. [Che24]

2.3 Digital Audio Rendering

Creating immersive sound plays an essential role in virtualizing systems making users feel more present in a given space. Through the years, several different approaches have developed to achieve this goal in various fields or applications.

2.3.1 Ambisonics

Ambisonics capture and play audio in spherical form. On one hand audio can be captured from all directions via sound field microphones and played back dynamically. This allows the sound to be adjusted to match the user's orientation and is thus able to produce realistic audio outputs of complex sound environments. On the other hand, ambisonics can be mixed, or created in a virtual environment. Many Digital Audio Workstations (DAWs) already come with extensions to create or manipulate ambisonic sound. Playback however is limited to either a matching number of speakers or has to be decoded dynamically to match headphones, or the given amount of output channels.[Art23]

In general, spherical harmonic functions are used to describe the sound-fields for ambisonics. These harmonics are represented as a hierarchical set of basis functions that are orthogonal to the surface of a sphere. Ambisonic channels can vary in order and degree. The order of an ambisonic system stands for the highest possible degree of an ambisonic channel. Zero-order ambisonics thus only have one channel, representing an omnidirectional sound pressure field. First-order ambisonics adds three spherical harmonics of degree one. These represent the expansion of the sound field surrounding the listener and are often used to encapsulate three dimensional sound. Lastly, higher-order ambisonics (HOA) are able to provide more realistic representations of the sound field, especially sound origins are more accurately represented. For this, HOA use higher spherical harmonic channel degrees. The downsides of HOAs in comparison to first-order ambisonics is cost and processing and playback. A fourth order ambisonics sound field should for example be represented as 25 evenly distributed speakers. To combat this downside, several solutions have solidified over the years, allowing HOA recordings to be accurately down-scaled to match a audio setup with lesser speakers than normally required. In doing so, the quality of immersion also gets reduced however.[GKSP24]

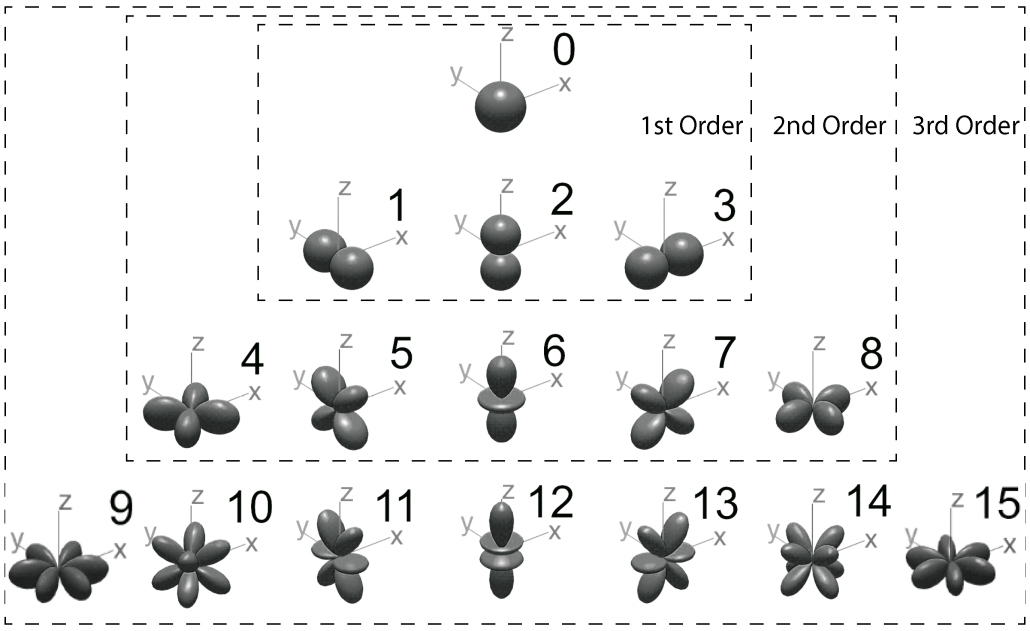

Figure 2.1: This graphic shows the first few orders of ambisonics. Each subsequent order incorporates all the previous ones, which makes ambisonics scale quite quickly. Each of the channels cover the the sound field represented as spheres, which shows how much more accurate ambisonics become, when scaled up to a higher order. Image source : [NSA+20]

Decoding ambisonics to a specific preexisting speaker layout also comes with a few challenges, since the output channels for the audio must match their position in the given space. The All-round ambisonic decoding (ALLRAD) uses exactly this mapping approach before decoding the audio sample, making it very flexible in use. Another approach is energy-preserving decoding (EPAD). Here, the entire ambisonics sound field is transformed to represent incomplete spheres to match the speaker layout. A comparison presented in Zotter and Frank shows that for hemispherical or incomplete spherical loudspeaker arrays, ALLRAD is the superior method. It produces fewer directional errors, has greater flexibility and the implementation is simpler compared to EPAD.[FZ19]

Because of these attributes, ambisonics are extensively used in VR and AR environments. For example, omnidirectional videos (ODVs) can be recorded or created using ambisonics and then consumed by users with matching hard- and software. Being able to turn 360° as the viewer and focusing your senses in the direction of your choosing creates high levels of immersion. [ROS19] In addition ambisonics are often used to create large-scale augmented reality rooms that contain a large amount of speakers to accurately match requirements of HOAs. One example for these rooms is the SARC Sonic Lab at Queen's University Belfast, a room that is equipped with 48 speakers individually controllable speakers and different audio levels for increased immersion. Additionally, for research

purposes, more speakers could be installed, to increase the order of ambisonics. [Bel04]

2.3.2 Object-based Audio

Object-based audio encodes sound more dynamically. Different sound-objects are created, which are then given a position and trajectory in a virtual space through metadata. Through this extension to standard audio signals, object-based signals are not limited to, or hard coded for a specific number of speakers, contrary to traditional channel-based audio. Object-based audio can be set up for any speaker configuration, given the speaker positions are known. This innovation, however, increases the amount of data for audio projects by quite a large amount. [Jea19]

Furthermore, the specific layouts can be customized in a way to suit the users specific needs. Hearing impaired listeners for example could increase the difference between background noises and dialogue to better understand the spoken words. Football fans could change the streamed audio in such a manner that the sound more accurately represents a live stadium feeling. To take full advantage of this object-based approach, however, systems need to be able to incorporate the metadata produced by the object-based audio. For this several standards have solidified themselves in the current object-based audio research. MPEG-H for example, contains the transmission pipeline for audio objects. Multi-dimensional audio (MDA) is a metadata model that includes bit-stream representations. Typically, the metadata consists of position and spread as well as size or diffuseness.[CFF+18]

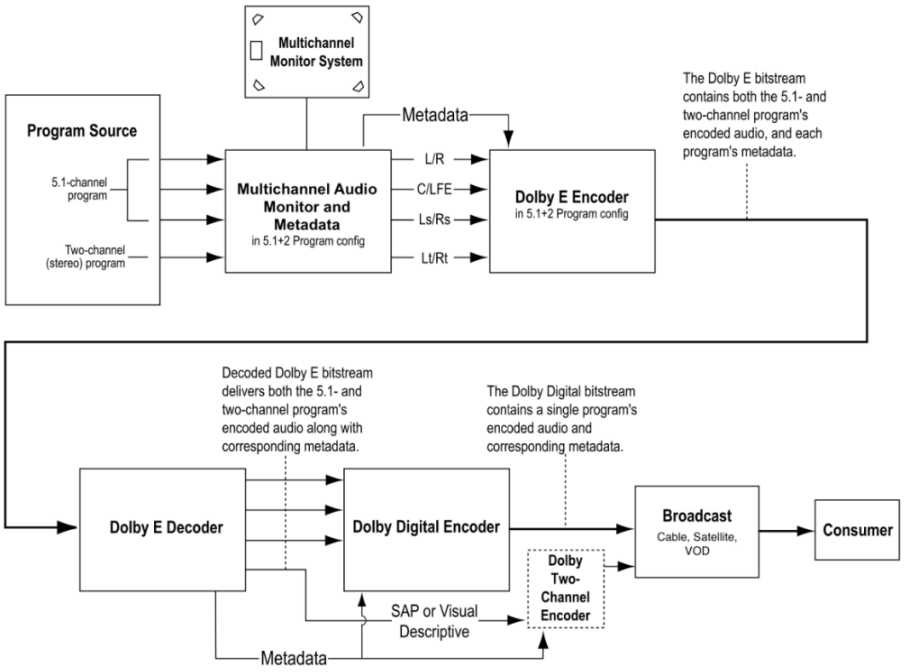

Figure 2.2: This is the Dolby Atmos metadata pipeline from production to consumer. [Dol22]

Because of this adaptability, streaming platforms like Netflix and Amazon Prime primarily use object-based audio formats like "Dolby Atmos" to achieve a level of immersion formerly only available to multi speaker cinema setups. This high-level sound system from the "Dolby" corporation is aimed towards cinema and television, but also extends into the music domain. [PC23] Originally premiering in 2012 in cinema Dolby Atmos, has been improved and extended upon regularly. Nowadays, there are a plethora of services and devices like speaker sets, that are supported by Dolby Atmos and can optimally decode the object-based metadata produced by Atmos Audio productions. Digital audio Workstations like the one used for this thesis, are also able to en- and decode sound in a way that Dolby Atmos compatible end-devices are able to decode. [Cab20]

The metadata for Dolby Atmos is carried together with the audio data through two regular digital audio channels (AES/EBU or S/PDIF) and consists of informational as well as control meta data. Informational metadata consists of parameters that do not affect the bit stream or the decoding process. Examples of informational data are copyright bits, or room types. The control metadata are all the necessary transformation information for the en- and decoder, like low-pass filters or down-mixing information. One of the most important control parameters is the dialogue level, representing the average loudness of dialogue in the given audio recording. In doing so the entire audio becomes matched between program sources and enables dynamic range control profiles

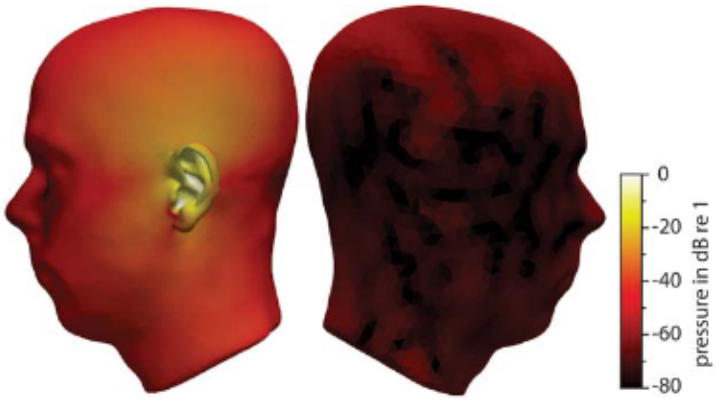

Figure 2.3: Example of the color-coded sound pressure effect on a model 3D model head at 7kHz in decibels. This figure represents a step in the Mesh2HRTF open source project, that calculates personalized HRTFs based on virtual meshes. Additionally to the head and pinna shown here, the torso is also included in relevant calculations. [FWJ+23]

for consumers with less than optimal listening-environments. This also means that audio without dialogue, still requires and depends upon the correct setting of the dialogue level. [Dol22]

2.3.3 Binaural Audio

Lastly Binaural Audio aims to enhance user immersion without a significant increase or change in hardware. For binaural audio to function properly, all the users need are headphones with stereo capabilities. This makes it extremely accessible for all kinds of users. Binaural Audio rendering uses the human biological hearing processes to its advantage by simulating a virtual audio location through acoustic cues, like interaural time differences and sound distortions created by the auricle. The main part that allows this rendering is the head-related transfer function (HRTF). The HRTF changes sounds depending on their location relative to the hearer's eardrum and incorporates how sound would change through the diffraction and reflection of our physical body. [TT23]

Since every human has a different set of physical features, personalized HRTFs would make for an optimal sound conversion and thus immersion. Normally personalized HRTFs are calculated based on measuring in an anechoic chamber, where the listener is positioned in the center and equipped with in-ear microphones. Then, speakers play reference signals in a spherical array, covering all possible azimuth and elevation angles. Since this method is quite time intensive and costly, other approaches have been developed like rotating the users head in front of a single loudspeaker and gathering data through a head tracking device.[JKSS23]

Another broad HRTF calculation method is acoustic simulation based on three-dimensional

body scans. Here, 3D mesh data are taken to simulate the diffraction of sound waves through the head and ear. Another approach is to calculate the HRTF through point cloud of the head, reducing calculation time and workload. Although these methods are more accessible than traditional HRTF calculations, they still require time consuming and reliable personalized data.[JKSS23]

Thus, generic HRTFs have been used frequently, especially in commercial environments with greater amounts of users. To generate generic binaural data, large HRTF datasets with associated head and ear measurements are utilized. These can then be either combined using averages or selected based on user input to quickly generate a satisfying binauralized output. Recently , deep learning approaches have attempted to tailor HRTFs to users based on visual input, these solutions however suffer from the same shortcomings as the acoustic simulations.[JKSS23]

As mentioned in 2.1, binaural audio is already used in standard digital games. However, this technology is also being extensively tested and applied in virtual reality applications as well as music and video production. Here, the most frequent usage of HRTFs is the average generic HRTF, since it is the least costly, and immersion suffers comparatively little when compared to personalized traditional HRTF calculations.[ATMK18] Additionally, since binaural audio focuses mainly on data output, it is often adopted by or used in conjunction with object-based audio or during down sampling of ambisonics. [PC23]

For this thesis a simplified version of object-based audio was used, that solely consists of azimuth and elevation values as metadata, which is then used during the binaurlization process to define the position of individual sound sources.

CHAPTER 3

Methodology

This chapter will go over the design of the system and give details about the choices of applied technologies. The main parts of the prototype, its structure and individual approaches to selected processes like binauralization are discussed here.

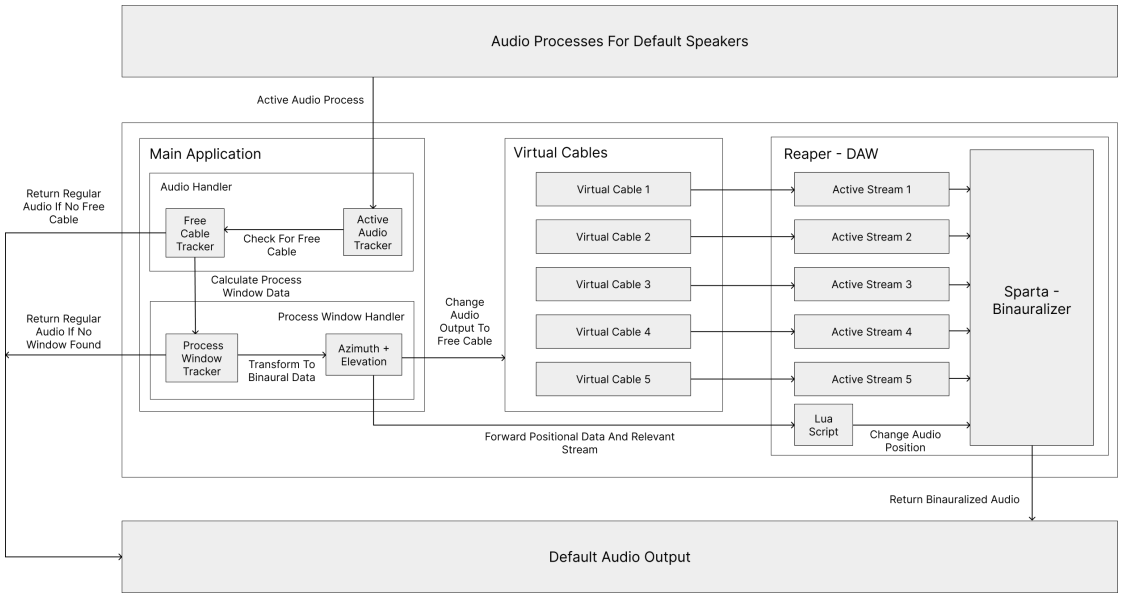

Figure 3.1: Schematic overview of how the prototype binauralizes audio processes and its general Layout. This schematic also shows how audio and positional data flows through the different segments of the application.

3.1 System Design

The desktop application for this thesis includes three main components. First, there is the central application that does the setup, most transformations, and the cleanup once the application is closed. It is the most integral aspect of the application and delegates tasks to all the other working parts. Then, there are virtual audio cables that single out audio objects of different processes, which allow individual audio instances to be given different virtual positions. Lastly, there is the Digital Audio Workstation (DAW). Through it, the audio is streamed and then binauralized. Afterwards it is returned to the standard output speakers.

3.1.1 Binauralization Approach

This prototype aims to combine two-dimensional objects, the process windows on screen, and three dimensional sound objects. The union between sound and window position allows for different approaches and variations. For this project, every active and available screen is combined into one grid and interpreted as a hemisphere. This interpretation allows for process windows to be mapped precisely for the binauralization process since during the transformation, spherical coordinates are used. This means that windows located on the left side of the monitor would produce sound that is coming from the left side of the user's head. Similarly, windows that are located at the top of the screen would therefore be mapped right above the user's head. Since sounds need not be mapped as rectangular shapes, the positional data of the center point of a window is taken.

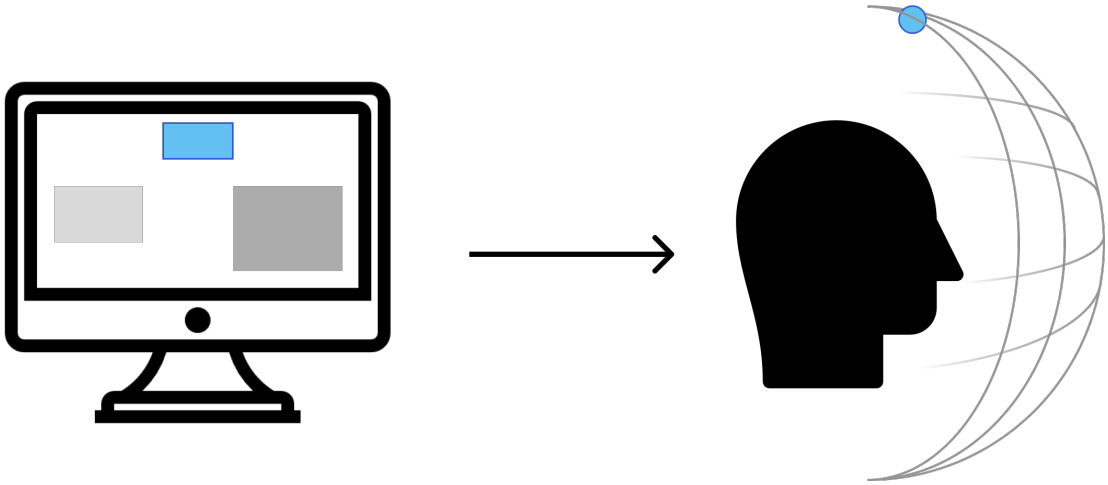

Figure 3.2: Simplified representation as to where an audio producing tab would be placed on the virtual hemisphere. The audio object (blue) gets converted from a rectangle into a single point on the hemisphere. Screen and Head Icon were taken from Flaticon.com.

In order to also include the other half of the spherical virtual space around the user and to increase the amount of immersion, windows that are minimized, are mapped behind

the listener to create an audible correlation to that action. For this the location data is simply inverted.

The center point of the screens always correlates to the center of the frontal hemisphere. Should multiple screens be used while running the prototype, the operation system returns a two-dimensional rectangle that is the sum of the screens measurements. This sum is then mapped onto the hemisphere, which may sometimes lead to counter intuitive center points. This limitation however, will be discussed later.

3.1.2 Virtual Cables

Necessary features of this project are virtual cables. These software applications emulate physical audio cables by enabling the transfer of audio signals without physical in- and output plugs. Instead for each cable installed virtual in- and outputs are registered for your computer. Once installed the respective ports should simply appear in the audio manager and can be accessed like any other audio port.

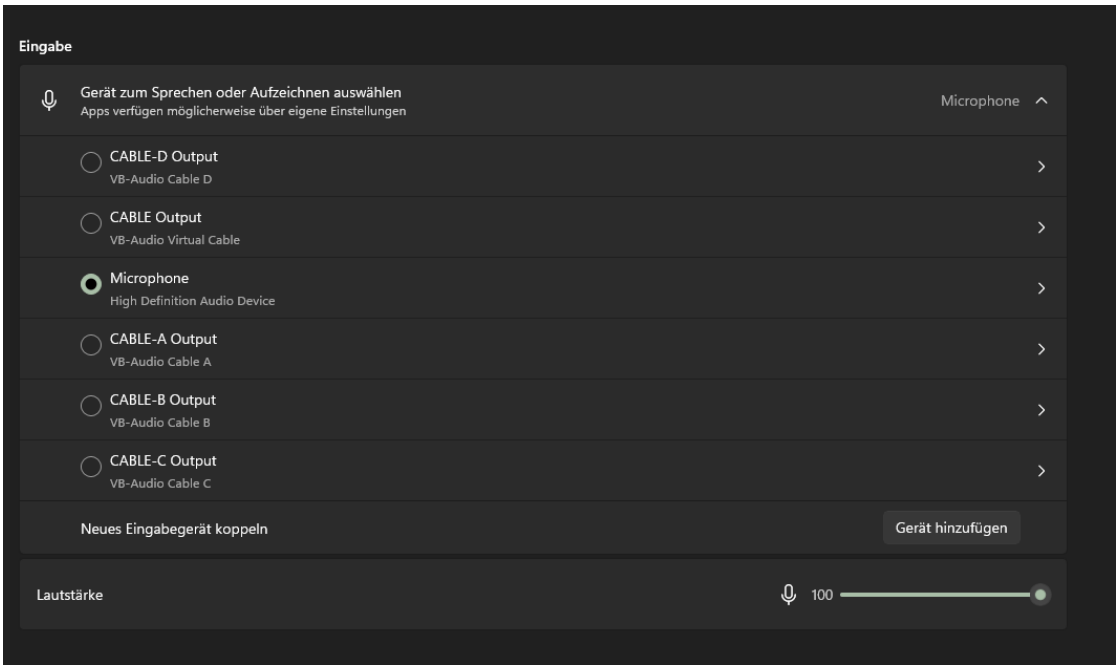

Figure 3.3: This image shows the list of all available audio input ports, once the cables have been installed.

In this application, the virtual cables act as a transmitter to the DAW and allow sound to be individually edited. This means that each audio cable represents a possible audio object. The program itself is therefore limited by the number of cables available. For this project five virtual audio cables were installed, meaning that at any point in time the program can individually calculate the position of five different audio processes. The rest are simply ignored by the program and put out to the default speakers. For that

reason, every cable is constantly tracked by the main application. Free cables are put into a queue to await a new audio object.

3.1.3 Main Application

The main application is a program written with C# as its code base. Upon starting the program, a form is initialized, that gives the user information about current active windows their names and their positions, as well as what cable they are assigned to. Afterwards different DAW projects are started up and activated, more on that in chapter 3.1.4. Additionally, a loop is initialized that listens for active audio tabs that would be audible through the headphones of the user. To do so, all active audio emitting processes, that would normally be put out to the default audio endpoint, are listed. The previously mentioned, already running DAW projects, also have their audio output set to the default speakers, however these need to be ignored, in order to prevent loops or fragments. Simultaneously, another loop checks if the Virtual Audio Cables are currently in use or not and marks free cables.

As soon as an active audio session is found, the position, size, and name of the audio emitting tab needs to be calculated. To access these sets of data, platform invocation services (PInvokes) were used. This feature allows managed code, like this programs C# to access the native code of the machine. Then, since many processes may have a different sub-process for audio output and for managing their windows, the program runs through all open windows and searches for a matching name. If a matching window is found, the search is stopped and the window data gets calculated, otherwise the program will not be able to give information on the window, which causes the entire audio object to be left untouched by the binauralization process.

Process windows are depicted as four numeric values, or two points on a plane. The first two numbers depict the position of the upper left corner of the window on the screen, while the last two numbers denote the lower right corner. From this the width and height can be calculated. Since every sound object must be represented as a point on the virtual hemisphere, each window is represented through its center point. Furthermore, the binauralizer, to which this data is to be transmitted, does not use flat location data, but a combination of two different angles. For this, the relative positional data of the previously calculated center point needs to be gathered.

The sent window data consists of azimuth and elevation, two angles in a spherical space. Azimuth defines the horizontal orientation of the sound and elevation the vertical. Combining these two values allows any point on a sphere to be localized. Additionally, should the tab be minimized, the audio position will be inverted horizontally and therefore be positioned behind the user.

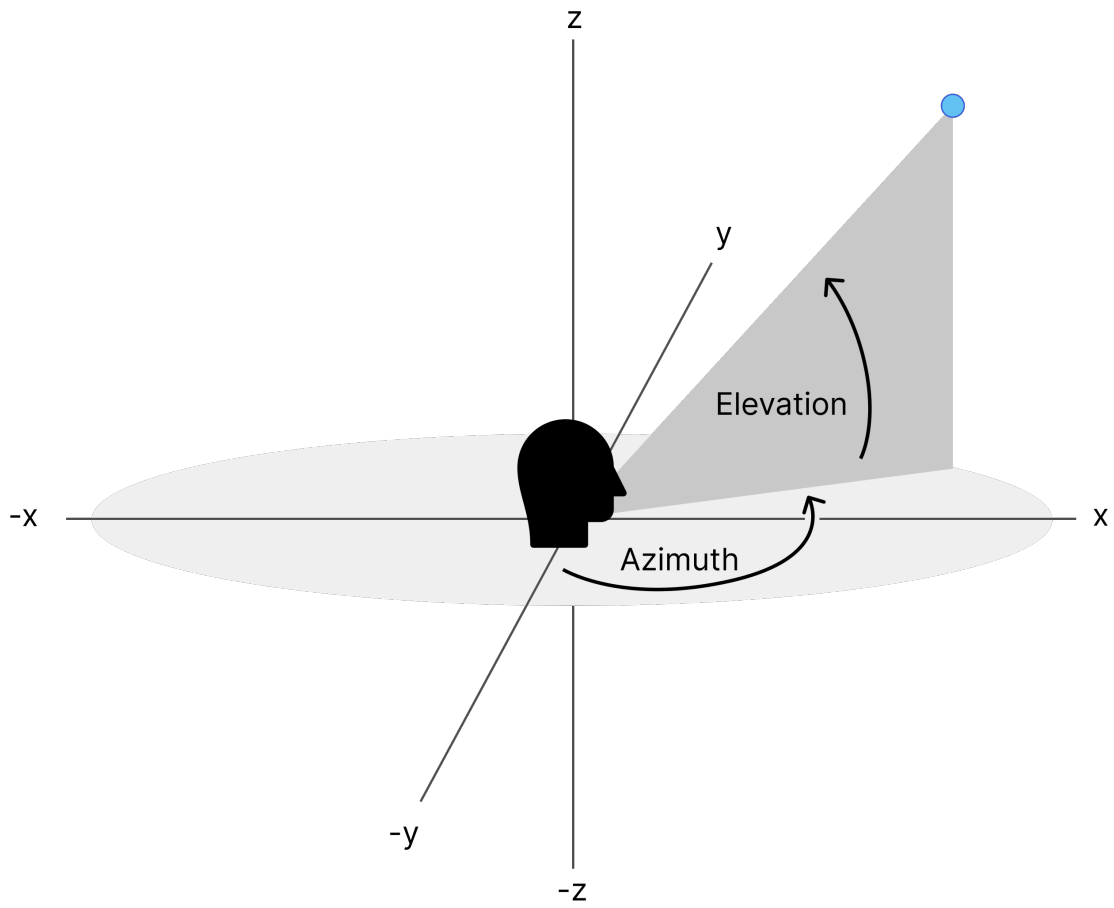

Figure 3.4: This image depicts how azimuth and elevation affect the orientation of a point (blue) on a spherical plane.

After gathering an active audio process with a designated window, the program will transfer the process to the next free cable or ignore it should all cables already be in use. In doing so, the process vanishes from the default audio output search. This audio delegation is done externally, via SoundVolumeView, [Sof25a] a program that specializes in sound management, that also allows the user to change audio outputs of specific tabs via command line. Thus, a command script is generated and then called through the application to set the current audio source to a previously selected, free cable.

3.1.4 Digital Audio Workstation

Digital audio workstations (DAW) are software programs that specialize in recording, editing, and streaming audio files. These have a plethora of plugin suits and other support which allow for complex audio editing. Furthermore, through the ability to script certain processes within the application and thus automating simple tasks, it is possible to allow

communication between DAWS and the main application of the prototype. The digital audio workstation used for this prototype is equipped with a plugin suite that allows encoding in binaural sound, SPARTA [oAAL25], and another plugin that allows scripts to be run on startup, SWS [Stu25].

Through the digital audio workstation, in this case Reaper [Inc25b], a number of projects are in a constant streaming state. For each integrated virtual cable, one project of Reaper is opened and set to listen to the output of one specific virtual cable. These Reaper projects themselves are streaming their audio to the standard speaker output, thus rejoining the binauralized strands with the rest of the unedited audio.

The second part of the Reaper application is a Lua script, the native language for Reaper. It is executed on startup and reactivates itself as well as another, data changing script in a constant loop. Through the main application the second script receives the name of the project that needs to change and the values of azimuth and elevation. These are then given to the matching project and its parameters of the binauralizer.

The binauralizer used here is from the SPARTA plugin, that offers encoding as well as decoding options for spatial audio approaches like binauralization. The so-called SPARTA-Binauralizer then takes the aforementioned azimuth and elevation values, calculates the position, and returns the final sound.

3.2 Choice Of Technology

3.2.1 Coding Environment and Used Language

The main application for this prototype is written with C# through Visual Studio [Cor25b]. This language offers useful libraries and is well supported and documented. Through C# the prototype could also be easily adapted to other systems like macOS or Linux. Visual Studio makes coding and debugging more efficient and offers a lot of tools for data visualization.

3.2.2 Virtual Cables

For virtually routing the different audio producing windows, VB-Cable (Virtual Audio Cable) [Sof25b] offered a satisfying performance for a low price point. The installation and configuration requirements are relatively low, and its usability and support makes VB-Cables stand out from its competitors. VB-Cables also offer great scalability, since every additional cable can be easily downloaded and then integrated into the existing program to add more maximum audio objects. For this project, the current amount of differently positioned audio tabs is 5, correlating to 5 Audio-Cables.

3.2.3 Audio Management Software

SoundVolumeView is a lightweight utility software developed by NirSoft. It allows users to manage and control sound-related settings on Windows and provides a centralized interface

for viewing and setting audio-related parameters. More importantly, SoundVolumeView offers a wide variety of well documented command-line functions, making it ideal to script with. Through this, the application can rapidly and reliably change the in- and output setting of individual processes, requiring only the name of the in- and output device as well as the process id.

3.2.4 Digital Audio Workstation

The Digital Audio Workstation is the second most crucial segment of the prototype. Here, Reaper (Rapid Environment for Audio Production, Engineering, and Recording), a comparatively cheap yet popular and powerful audio workstation, is responsible for finalizing the spatialization process and also for recombining the different audio objects back into the standard audio output. This is done through inherent capabilities that reaper offers, like accessible scripting and allowing for multiple projects to be opened at the same time. Additionally, through free extensions like SWS (Standing Water Studios) and SPARTA, more difficult actions are able to be executed. SWS specializes in adding advanced functionalities like running scripts on startup, which was used in this prototype. SPARTA adds spatial audio features that build the core of this project. For this, many different add-ons were looked at, however SPARTA offered the most ease of use and had very satisfying outputs. Furthermore, the SPARTA-Binauralizer allowed location data to be changed through a Reaper script.

CHAPTER 4

Implementation and Use

This chapter goes into detail about the implementation of the prototype and the different components. All required installations and setup actions are described here as well. Also, instructions to scale this project up or down in terms of audio objects are given in this chapter.

4.1 Used Assets, Packages and Environments

4.1.1 Windows 11

Since the tab recognition function and the auxiliary programs and plugins may be incompatible with other operating systems (OS), it is important to state that this prototype was created for and only tested on Windows 11. Specifically for the configuration Windows 11 Pro, Version 23H2. It is to be expected that the audio and window search process, that utilizes the inherent process structures of Windows 11, needs to be reconfigured on changing the operating system. Furthermore, older and no longer supported versions of the Windows operating systems might not be able to install the current Reaper Version along with its plugins.

4.1.2 Visual Studio

The Main Application is written on Microsoft Visual Studio Community 2022 with the version 17.10.0. Though it is possible to extend and change aspects in any C# coding environment. Furthermore, the program uses a variety of packages. For Audio recognition, the NAudio package version 2.2.1.0 was used. Here the MMDeviceEnumerator allows us to check relevant audio endpoints for active audio signals and give us a process id should one be found. Additionally, Windows Forms, Version 8.0.0.0 was used to create a simple window, that reflects project data like active Audio sessions and their positions. Through this form the program can also be shut down.

4.1.3 VB-Audio Virtual Cable

As previously mentioned, the virtual cables that connect audio objects to the digital audio workstation and therefore to the binauralizer, are from VB-Audio Software. In total 3 packages were used by this supplier. First, the standalone 'VB-CABLE Virtual Audio Device' running the 'Pack 43' configuration is an audio driver, that users can try out for free and includes one fully functional virtual cable. Additionally, for this program the auxiliary cables 'VB-Cables A+B Pack 43' and 'VB-Cables C+D version 2.1.5.2', were added to create a total of five maximum audio object channels. Once installed these cables simply appear in your audio in and output selections making their usage extremely intuitive.

4.1.4 SoundVolumeView

'SoundVolumeView' is an essential audio management software that gives not only a broad insight into the current audio in- and outputs as well as running audio processes but also offers rerouting capabilities of audio outputs via command line. For this build the version 'v2.46' was used.

4.1.5 Reaper

The Digital Audio Station that manages the audio stream and does the binaural calculation, Reaper from Cockos Incorporated, runs the version v7.12 and was used with a license for small businesses or personal use. However, Reaper also offers a free trial in which the essential functionalities, for this project at least, are all covered. Added onto Reaper are two essential plugins that extend the functionality of the program for this individual project. First up, the SWS/S&M extension version 2.12.1.3 was installed. This open-source plugin aims to improve Reapers usability while also adding a bunch of additional features. Lastly, SPARTA, short for Spatial Audio Real-Time Applications, version 1.7.1. was installed and used through Reaper. It is a collection of open-source VST/LV2 audio plug-ins that contain a binauralizer as well as several other spatial audio en- and decoders. Both plugins need to be installed and connected to Reaper. Without them the DAW would not be able to transform the ingoing audio streams into flexible binaural outputs.

4.2 Reaper Project Setup

After acquiring all aforementioned applications and assets, all parts need to be installed, connected and set up properly. First, after downloading Reaper and its plugins, it is vital to connect the SWS and SPARTA extensions with your DAW. After unpacking the folders and running the installers, all extensions need to be activated in Reaper. Under Preferences and VST, short for virtual studio technology, Reaper asks the user to add the plugin directory paths to a list. Afterwards Reaper needs to re-scan for all plugins

and clear its previous cache to ensure that the new extension is detected. After installing everything correctly, the added features should be visible in their planned location.

Next, a running audio stream is required. To do that, the input and output devices need to be set. Normally, the output endpoint of the stream should always be the standard speakers, however it is entirely possible to set this output to a specific device. Keep in mind that changing this after setup completion may take time. For the input each project needs to be set to a different virtual cable output. For the specific audio setup see figure 4.1.

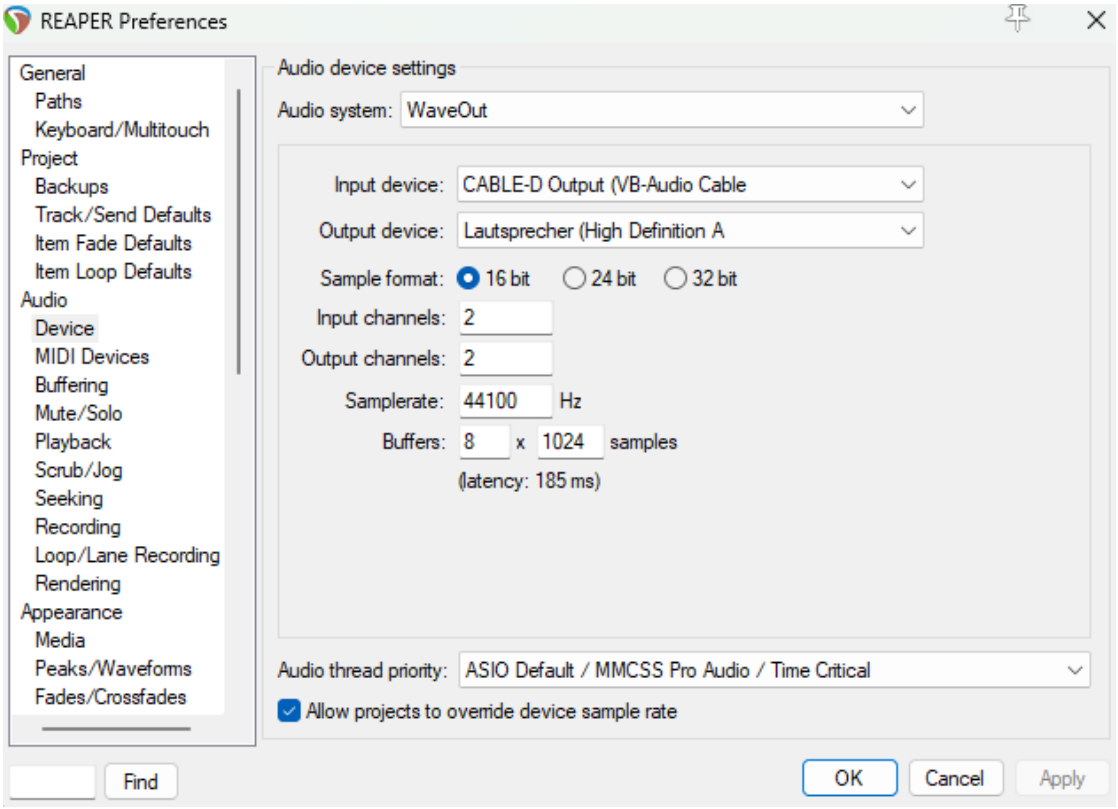

Figure 4.1: This Image shows the exact audio setup of the Reaper stream for the fifth Cable called CABLE-D. The output is set to the standard output speaker.

Afterwards, create a new audio track and set the input to your standard Input 1. This means that this audio channel is listening to mono inputs from the previously set cable. Then, click the record button to activate this track. After setting up the audio stream, add the SPARTA binauralizer to the master track, which handles the output of the stream. This is done by clicking on the FX-button, as seen in figure 4.2 and selecting the AALTO: sparta_binauralizer. After activating, the user interface for the binauralization pops up. While being a very informative visual representation of azimuth and elevation values, this interface may be closed. If the effect was correctly activated, the button should now be turquoise in color.

Lastly head to the action list and add two lua scripts, the change-binaural-script, that can be extracted from the main application and the main-script. This main-script's sole task is to continuously reactivate itself and calling the change-binaural-script. To make this script active on startup use the SWS startup action feature that can be found under extensions. After finishing, save this project under a folder of your choice and then repeat this process for every virtual cable that you want to include in your program, changing only the input device and name of the project.

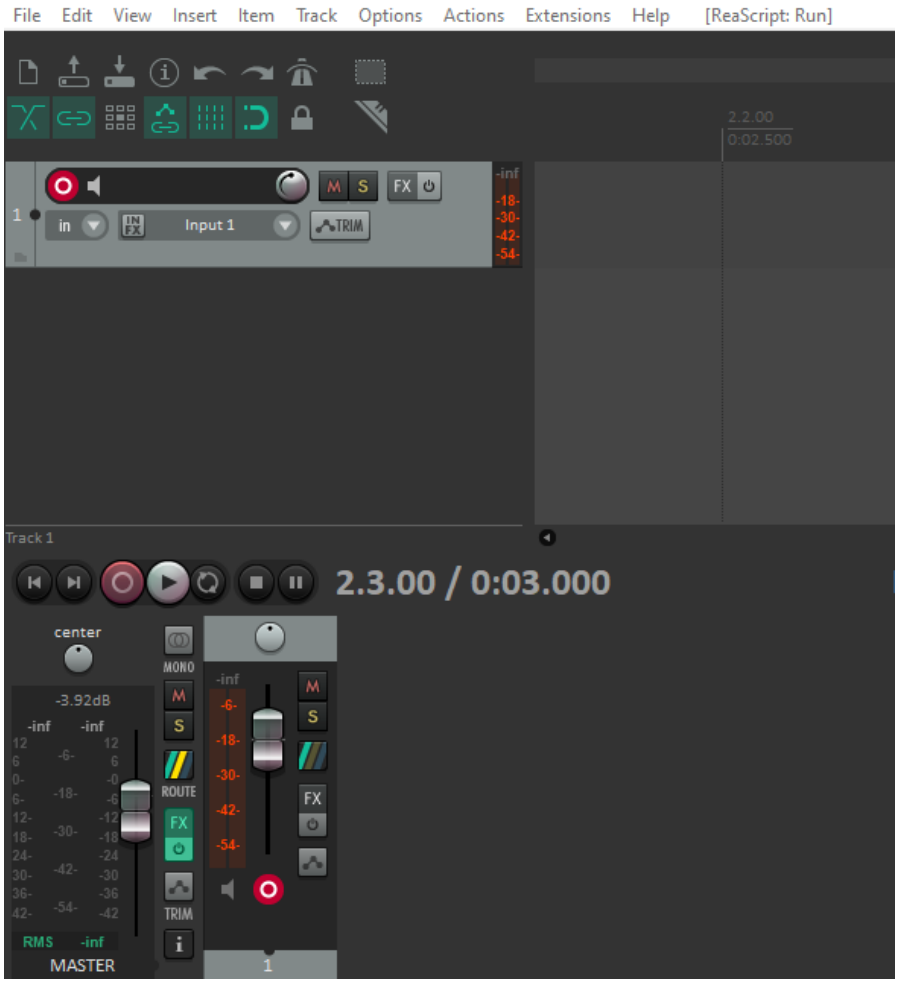

Figure 4.2: Here, a correctly set up Reaper stream is shown. On the top left, an active stream is listening to incoming sound from a set cable. In the top right corner, the text 'ReaScript: Run' informs the user, that a script is active. The bottom left track, titled MASTER, has an active FX-button, signaling that the binauralizer is active for this stream.

4.3 Interaction Flow

This project is meant to be run alongside other programs on your computer that require your attention. Therefore, the user interface and the interaction requirements are kept to a minimum. Once correctly started up, this prototype requires basically no additional input from the user, as all the routing is done automatically and the controls function indirectly through moving the tab around the screen or minimizing it. To gain visual information on one of the audio objects, it is possible to open one of the minimized Reaper projects and view the local binauralization interface.

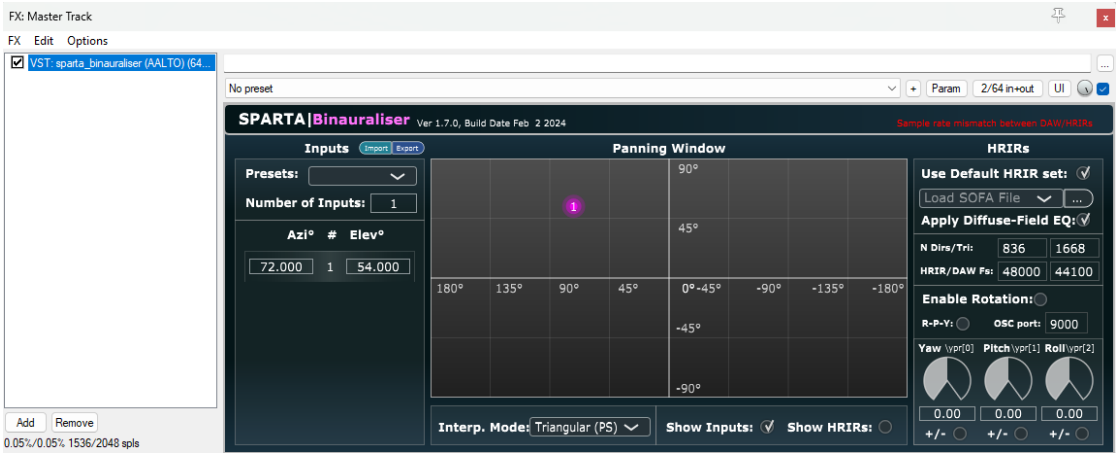

Figure 4.3: This image shows SPARTA's native representation of the binauralization process. Left are the current azimuth and elevation angels. The pink dot in the center represents the audio objects current location. The grid consists of the azimuth (horizontal line) and elevation (vertical line) values as well.

As already mentioned, it is also possible to in- or decrease the number of virtual cables and therefore the amount of individual audio objects. Generally, users have 1 to 2 open sound producing tabs and rarely reach the 5 maximum cables that are installed for this prototype. Should one want to put this technology to the limit, it is required to install additional audio cables and then add an additional reaper project like explained above with its input set to that cable's output. Lastly the main application needs to be extended.

4.4 Limitations and Known Issues

This prototype was designed as a first look into what an application would feel and sound like and is therefore not optimized. For future iterations or updates several aspects could be looked at and improved. First, the routing of audio tabs posed a great challenge. The general flow of the prototype searches for active audio tabs, then gathers the internal process id to then search all open windows for one with a matching process id and

title. This is important since applications like Chrome have many different sub-processes running at the same time, all being named differently.

Programs like internet browsers allow multiple windows to be opened at once, which confuses the title matching algorithm. For now, it is not possible for the program to differentiate between different windows of the same browser and takes the position of the last active window. Should one browser produce multiple audio sources, they are bundled together and streamed through a single cable. Another limitation is the usage of Reaper or any other auxiliary DAW for that matter. Since the prototype uses Reaper for its binauralization calculation, it needs to be excluded from being caught and binauralized itself. For that, an exclusion option was added to the prototype. Here users can add processes that are ignored by the program and therefore will not be binauralized.

Processes that have in- and output audio, like Zoom [ZVC25] or Skype [Cor25a] do not function properly as of yet. The users Audio input vanishes and the listener's incoming audio does not get caught by the active audio outputs, though the prototype only specifically looks for output audio and is also limited to elusively look at output audio. Communication programs like Discord [Inc25a], that also allow videos to be played directly in the application, work well, only as long as no call is started. This problem was tested and persisted on all the aforementioned programs. Solving this issue would open the prototype up to a whole new array of usages. This issue could stem from an access restriction on call audio in- and outputs.

Lastly, the performance, though acceptable and not the biggest focus of this prototype, could be improved immensely, should the binauralization happen within the main part of the program. Without requiring Reaper, Scripts, command line calls and additional extensions the entire binauralization process could be sped up to increase response time and decrease data flow and startup time.

CHAPTER 5

Results and Outlook

For this project, a prototype application that binauralizes outgoing process sound in regard to its window position on screen was created. The application functions for most windowed audio extruding computer processes and matches outgoing sound with current location on screen. For this, the screen forms a virtual sphere around the user. Processes that produce sound and correspond to a specific window, are captured and streamed through a binauralizer, which changes the position of the sound to correlate to the location of the process on screen. In the prototype, up to five different sound objects can be individually binauralized while other programs are audible in a standard stereo fashion. Would the user change the size or the position of an audio producing window, the perceived origin of sound switches up accordingly in real time, achieving a greater sense of immersion for common office tasks, while using regular and cheap hard- and software. Visit `https://www.vr.tuwien.ac.at/projects/audio-spatialization-of-windows/` to listen to example videos or to check out the code base.

5.1 Future Improvements

This being a prototype and functioning as a test to what could be possible in the realm of audio binauralization for two-dimensional computer screens, many aspects are open to improvements. First, the startup time and latency are not yet optimized and could improve usability and immersion drastically. A big reason for that is that the application accesses a digital audio workstation for its calculations and requires multiple audio streams. Delegating these calculations and data flows has a substantial impact on how fast the program can activate and react to change.

Regarding sound quality and aural immersion, another improvement could be made. For now, the prototype only calculates the incoming sound of processes through a single channel and maps the window onto a single location. However, most processes create stereo sound that already offer some sort of immersion within the process itself. These

nuances are getting lost during the binauralization process of this project. To binauralize a stereo-stream, both channels need to be mapped simultaniously in a chosen proximity to each other. The listener would then be able to differentiate sound positions within a single binauralized process.

Another closely related topic would be proximity. More costly and optimized binauralizers offer the inclusion of proximity and distance calculation, through changes in amplitude and frequency, as well as inter-aural level differences (ILD) and inter-aural time differences (ITD).[PSS21] With this feature the application could implement proximity and distances to processes through the size of their windows. Large, nearly full-screen windows could be calculated to sound closer to the user and small windows would be perceived as further away, increasing overall immersion by another dimension.

Next, the interface, that shows what audio objects the program has picked up and where they are currently being streamed, as well as their window position, is not optimized and could also be used for additional visual information. Reaper through the SPARTA plugin for example offers great insight into where the binauralized object is currently located virtually - See figure 4.3. In a similar fashion, all of the virtual objects for this prototype could be displayed simultaneously through a small overlay to further understanding and ease of use.

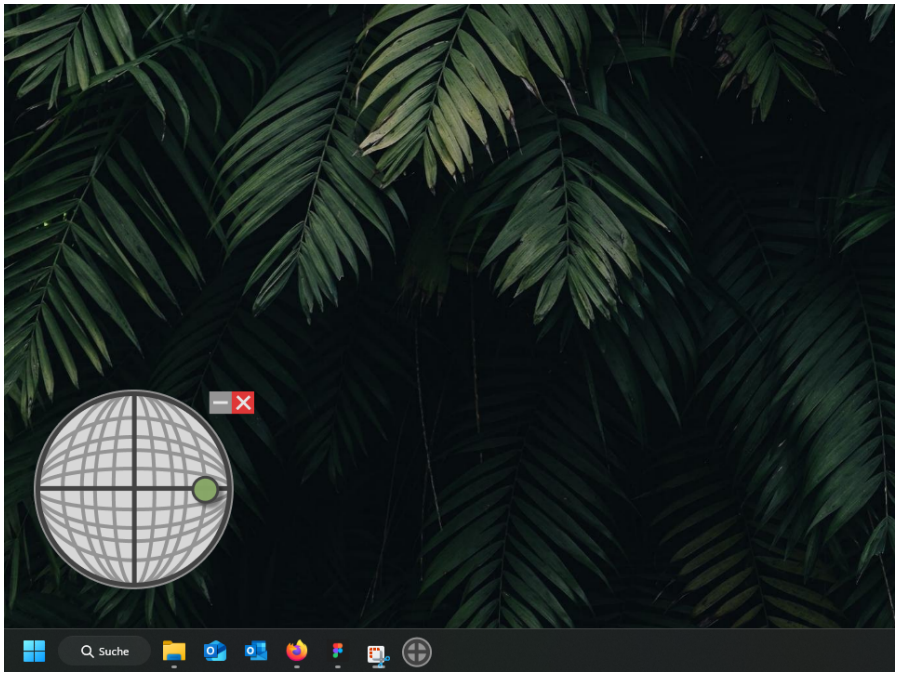

Figure 5.1: This mockup shows what a potential overlay for this project may look like. Users would be able to drag the visualization into a preferred location on their screen and scale its size to their liking. The green dot on this mock-up would symbolize the three dimensional location of one binauralized window.

Lastly, users often have their screens set up in a specific and personalized fashion. Some, for example, focus mainly on one center screen while having another auxiliary one to the left side of the main screen. This would then change the required mapping of the frontal hemisphere along with its center point. Could the program include those preferences, a much more immersive and realistic virtual audio environment could be calculated before the binauralization. This could be done through a separate setup- or options-window, that may be accessed during runtime in which users are able to freely change the center point of their hemisphere.

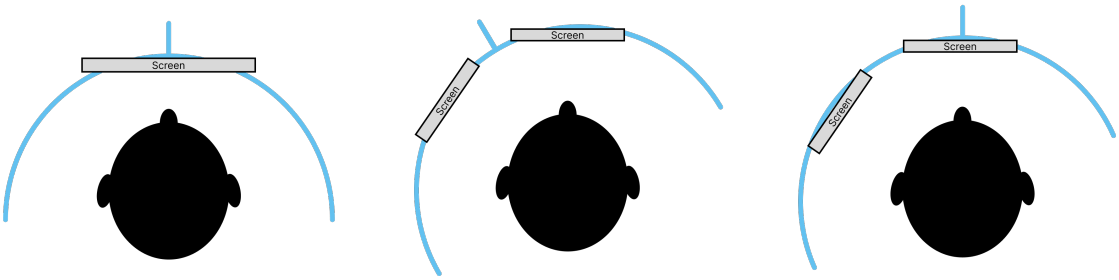

Figure 5.2: This figure shows how the center of all active screens gets calculated. The left image shows the orientation of a standard base case with a frontal single monitor. The image in the middle represents the current orientation of the hemisphere, should the user have a main monitor and an auxiliary secondary monitor to the side. Here, the center-point of the hemisphere would not be directly in front of the user, leading to a slight misalignment of seen and heard processes. On the right side, a corrected version of this example is shown, explaining how the hemisphere must be warped to create optimal immersion.

5.2 Extended Applications

Generally, the software was created to aid users in mundane computer work processes, however, should this project extend its availability and improve on critical usability aspects, this concept could be introduced into a variety of professional settings. First, workers in an audio centric jobs like sound production often come into contact with a plethora of simultaneously playing sounds. Through the binauraliztion process this workload could be virtually spread out to decrease workload. Broadcast technicians or post production specialists also naturally have multiple incoming sounds to work with and thus would also benefit form this project. Next, once communication software is available for the prototype, the immersion of e.g. Zoom calls could be increased greatly, helping office forces. Through additional extensions it might even be possible to treat different participants as single audio objects and give them their individual binauralized position, further increasing the overall usability.

5.3 Conclusion

This prototype demonstrates the potential to significantly enhance the sense of immersion in standard digital experiences by transforming the base stereo output of a computer into multiple binaural audio objects that reflect their perceived position on screen. This approach achieves a more intuitive, immersive and spatially aware auditory experience without relying on expensive or isolating hardware, such as VR goggles or head-mounted displays. By leveraging entry-level applications and simple headphones, this method opens the door for a broader audience to access immersive technologies. This approach could change the way operation systems handle sound to improve workflow, reduce overstimulation, and increase attention span for users and make working with multiple processes easier. To test out the prototype or to listen to example videos, head to `https://www.vr.tuwien.ac.at/projects/audio-spatialization-of-windows/`.

List of Figures

2.1 This graphic shows the first few orders of ambisonics. Each subsequent order incorporates all the previous ones, which makes ambisonics scale quite quickly. Each of the channels cover the the sound field represented as spheres, which shows how much more accurate ambisonics become, when scaled up to a higher order. Image source : [NSA+20] 8
2.2 This is the Dolby Atmos metadata pipeline from production to consumer. [Dol22] 10
2.3 Example of the color-coded sound pressure effect on a model 3D model head at 7kHz in decibels. This figure represents a step in the Mesh2HRTF open source project, that calculates personalized HRTFs based on virtual meshes. Additionally to the head and pinna shown here, the torso is also included in relevant calculations. [FWJ+23] 11

3.1 Schematic overview of how the prototype binauralizes audio processes and its general Layout. This schematic also shows how audio and positional data flows through the different segments of the application. 13
3.2 Simplified representation as to where an audio producing tab would be placed on the virtual hemisphere. The audio object (blue) gets converted from a rectangle into a single point on the hemisphere. Screen and Head Icon were taken from Flaticon.com. 14
3.3 This image shows the list of all available audio input ports, once the cables have been installed. 15
3.4 This image depicts how azimuth and elevation affect the orientation of a point (blue) on a spherical plane. 17

4.1 This Image shows the exact audio setup of the Reaper stream for the fifth Cable called CABLE-D. The output is set to the standard output speaker. 23
4.2 Here, a correctly set up Reaper stream is shown. On the top left, an active stream is listening to incoming sound from a set cable. In the top right corner, the text 'ReaScript: Run' informs the user, that a script is active. The bottom left track, titled MASTER, has an active FX-button, signaling that the binauralizer is active for this stream. 24

4.3 This image shows SPARTA's native representation of the binauralization process. Left are the current azimuth and elevation angels. The pink dot in the center represents the audio objects current location. The grid consists of the azimuth (horizontal line) and elevation (vertical line) values as well. . 25

5.1 This mockup shows what a potential overlay for this project may look like. Users would be able to drag the visualization into a preferred location on their screen and scale its size to their liking. The green dot on this mock-up would symbolize the three dimensional location of one binauralized window. 28

5.2 This figure shows how the center of all active screens gets calculated. The left image shows the orientation of a standard base case with a frontal single monitor. The image in the middle represents the current orientation of the hemisphere, should the user have a main monitor and an auxiliary secondary monitor to the side. Here, the center-point of the hemisphere would not be directly in front of the user, leading to a slight misalignment of seen and heard processes. On the right side, a corrected version of this example is shown, explaining how the hemisphere must be warped to create optimal immersion. 29

Bibliography

[Art23] Daniel Arteaga. Introduction to ambisonics, 2023. doi:10.5281/zenodo.7963105.

[ASB+19] Sarvesh Agrawal, Adèle Simon, Søren Bech, Klaus Bærentsen, and Søren Forchhammer. Defining immersion: Literature review and implications for research on immersive audiovisual experiences. *Journal of Audio Engineering Society*, 2019. doi:10.17743/jaes.2020.0039.

[ATMK18] Cal Armstrong, Lewis Thresh, Damian Murphy, and Gavin Kearney. A perceptual evaluation of individual and non-individual hrtfs: A case study of the sadie ii database. *Applied Sciences*, 8(11), 2018. doi:10.3390/app8112029.

[Bel04] Queen's University Belfast. Sarc sonic lab, 2004. Accessed: 2025-02-03. URL: https://visualstudio.microsoft.com.

[Blo18] Johan Blomberg. The semiotics of the game controller. *The International Journal Of Computer Game Research*, 18(2), 2018. ISSN:1604-7982.

[BM23] Patrick Bergsten and Kihan Mikita. Spatialisation of binaural audio with head tracking in first-person computer games, 202023. Accessed: 2024-11-04. URL: https://www.diva-portal.org/smash/get/diva2:1784453/FULLTEXT01.pdf.

[Cab20] Roberto Cabanillas. Mixing music in dolby atmos, 2020. Accessed: 2025-02-03. URL: https://digitalcommons.csumb.edu/caps_thes_all/956.

[CFF+18] Philip Coleman, Andreas Franck, Jon Francombe, Qingju Liu, Teofilo de Campos, Richard J. Hughes, Dylan Menzies, Marcos F. Simón Gálvez, Yan Tang, James Woodcock, Philip J. B. Jackson, Frank Melchior, Chris Pike, Filippo Maria Fazi, Trevor J. Cox, and Adrian Hilton. An audio-visual system for object-based audio: From recording to listening. *IEEE Transactions on Multimedia*, 20(8):1919–1931, 2018. doi:10.1109/TMM.2018.2794780.

[Che24] Andrew N. K. Chen. Information systems research of immersive technologies – virtual reality, augmented reality, and mixed reality. *Journal of Information*

Technology Case and Application Research, 26(3):256–263, 2024. doi:10.1080/15228053.2024.2401993.

[CNP23] José Manuel Cimadevilla, Raffaella Nori, and Laura Piccardi. Application of virtual reality in spatial memory. *Brain Sciences*, 13, 2023. doi:10.3390/brainsci13121621.

[Cor25a] Microsoft Corporation. Skype, 2025. Accessed: 2025-01-21. URL: https://www.skype.com.

[Cor25b] Microsoft Corporation. Visual studio, 2025. Accessed: 2025-01-21. URL: https://visualstudio.microsoft.com.

[Dol22] Dolby. A guide to dolby metadata, 2022. Accessed: 2025-02-03. URL: https://professionalsupport.dolby.com/s/article/A-Guide-to-Dolby-Metadata?language=en_US.

[FMMH19] Andrea Ferlini, Alessandro Montanari, Cecilia Mascolo, and Robert Harle. Head motion tracking through in-ear wearables, 2019. Accessed: 2024-11-04. URL: https://www.cl.cam.ac.uk/~cm542/papers/earcomp19.pdf.

[FWJ+23] Brinkmann Fabian, Kreuzer Wolfgang, Thomsen Jeffrey, Dombrovskis Sergejs, Pollack Katharina, Weinzierl Stefan, and Majdak Piotr. recent advances in an open software for numerical hrtf calculation. *journal of the audio engineering society*, 71:502–514, 2023.

[FZ19] Matthias Frank Franz Zotter. *Ambisonics - A Practical 3D Audio Theory for Recording, Studio Production, Sound Reinforcement, and Virtual Reality*. Springer Nature Switzerland AG, 2019. doi:10.1007/978-3-030-17207-7.

[GKSP24] Fotis Georgiou, Claudia Kawai, Beat Schäffer, and Reto Pieren. Replicating outdoor environments using vr and ambisonics: a methodology for accurate audio-visual recording, processing and reproduction. *Virtual Reality*, 28:111, 2024. doi:10.1007/s10055-024-01003-1.

[Hä20] Polina Häfner. Categorization of the benefits and limitations of immersive environments for education, 2020. doi:10.46354/i3m.2020.mas.020.

[Inc25a] Discord Inc. Discord, 2025. Accessed: 2025-01-21. URL: https://discord.com.

[Inc25b] Cockos Incorporated. Reaper, 2025. Accessed: 2025-01-21. URL: https://www.reaper.fm.

[Jea19] Philip Jackson and et al. Object-based audio rendering, 2019. Accessed: 2024-11-28. URL: https://arxiv.org/pdf/1708.07218.

[JKSS23] Vivek Jayaram, Ira Kemelmacher-Shlizerman, and Steven M. Seitz. Hrtf estimation in the wild. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 88, 2023. doi:10.1145/3586183.3606782.

[KG21] Park S. Kyung G. Interactive effects of display curvature radius and display size on visual search performance and visual fatigue. *Hum Factors*, 63(7):1182–1195, 2021. doi:10.1177/0018720820922717.

[NSA+20] Miroslaw Narbutt, Jan Skoglund, Andrew Allen, Michael Chinen, Dan Barry, and Andrew Hines. Ambiqual: Towards a quality metric for headphone rendered compressed ambisonic spatial audio. *Applied Sciences*, 10(9), 2020. URL: https://www.mdpi.com/2076-3417/10/9/3188, doi:10.3390/app10093188.

[oAAL25] University of Aalto Acoustics Lab. Sparta (spatial audio real-time applications), 2025. Accessed: 2025-01-21. URL: https://research.spa.aalto.fi/projects/sparta_vsts/.

[PC23] Edwin Pfanzagl-Cardone. *The Art and Science of 3D Audio Recording*. Springer Nature Switzerland AG, 2023. doi:10.1007/978-3-031-23046-2.

[PSS21] Kranti Kumar Parida, Siddharth Srivastava, and Gaurav Sharma. Beyond mono to binaural: Generating binaural audio from mono audio with depth and cross modal attention, 2021. Accessed: 2025-01-27. URL: https://arxiv.org/abs/2111.08046.

[ROS19] Aakanksha Rana, Cagri Ozcinar, , and Aljosa Smolic. Towards generating ambisonics using audio-visual cue for virtual reality, 2019. Accessed: 2024-11-05. URL: https://arxiv.org/pdf/1908.06752.

[Sla18] Mel Slater. Immersion and the illusion of presence in virtual reality. *British Journal of Psychology*, 109:431–433, 2018.

[Sof25a] Nir Sofer. Soundvolumeview, 2025. Accessed: 2025-01-21. URL: https://www.nirsoft.net/utils/sound_volume_view.html.

[Sof25b] VB-Audio Software. Vb-cable virtual audio cable, 2025. Accessed: 2025-01-21. URL: https://vb-audio.com/Cable/.

[SP18] Ayoung Suh and Jane Prophet. The state of immersive technology research: A literature analysis. *Computers in Human Behavior*, 86:77–90, 2018.

[SP22] Nicholas M. Santos and Alan Peslak. Immersive technologies: Benefits, timeframes, and obstacles. *Issues in Information Systems*, 23:170–184, 2022.

[SPW21] D. Bürger M. Naujoks L. F. Martin K. Petri S. Pastel, C. H. Chen and K. Witte. Spatial orientation in virtual environment compared to real-world. *Journal of Motor Behavior*, 53(6):693–706, 2021. doi:10.1080/00222895.2020.1843390.

[Stu25] Standing Water Studios. Sws extension for reaper, 2025. Accessed: 2025-01-21. URL: https://www.sws-extension.org.

[Sun21] Xuejing Sun. Immersive audio, capture, transport, and rendering: a review. *APSIPA Transactions on Signal and Information Processing*, 10:1–24, 2021.

[TH24] Tran Minh Tung and Doan Thi Thanh Huong. Immersive marketing measurement: How vr, ar, and aiare transforming customer engagement tracking. *Journal of Electrical Systems*, 20(7):3808–3818, 2024.

[TT23] Matteo Tomasetti and Luca Turchet. Playing with others using headphones: Musicians prefer binaural audio with head tracking over stereo. *IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS*, 53:501–511, 2023.

[VD] Inc. Virtual Desktop. Virtual desktop. Accessed: 2025-01-13. URL: https://example.com.

[ZVC25] Inc. Zoom Video Communications. Zoom, 2025. Accessed: 2025-01-21. URL: https://zoom.us.